

CORTEX USERS GROUP

T Gray, 1 Larkspur Drive, Featherstone, Wolverhampton, West Midland WV10 7TN.
E Serwa, 93 Long Knowle Lane, Wednesfield, Wolverhampton, West Midland WV11 1JG.
Tel No: T Gray 0902 729078, E. Serwa 0902 732659

CORTEX USER GROUP NEWSLETTER (MAY 1988)

Issue Number 18

CONTENTS

1. Index
2. Editorial
2. More on Canyon and Missile command
3. MDEX utilities Terminal enable/dissable
6. Monitor Command Extension
7. MDEX to CDOS file transfer
8. Othello
13. Mastermind
14. Compressed screen save
16. Adverts

Editorial

You may notice the better print quality in the Newsletter this month. That is because we have invested in one of the Royal Office Master 2000 daisy wheel printers available from Matmos Ltd for £119.00. Its very good value for money. The printer works well with the Cortex using centronics interface. I believe that a version is also available with serial interface. The only problem with the printer is that there is no english character set that gives a £ sign instead of the #. To get the £ you have to print "<1B5B>" from cortex basic. There is a modified version available from Aztech office supplies which substitutes the £ for the # but it is slightly more expensive. Also the ribbon supplied with the machine may be very short so it would be as well to order a spare one.

Matmos Ltd, 1 Church street, Chuckfield, West sussex.
0444 414484

Aztec Office Equipment Ltd "High winds" South End, Much Hadham,
Herts. 027 984 2036

Corrections for Canyon and Missile Command P.D.Wrightson

The problems that people are having with the games in the christmas issue due to the keyboard reading routine also reading other things like the disk size and density jumper settings and some device interrupt signals. The solution is not to read the extra information. The following modifications to the programmes cure the problem.

Canyon

```
5 BASE 16
670 A=KEY[0]:A=CRF[8]:IF A<>CRF[8] OR B=1 THEN GOTO 350
680 IF A=8 THEN P1=-1:GOTO 350
690 IF A=9 THEN P1=1:GOTO 350
700 IF A=52H AND PO>80 THEN P1=-80:GOTO 350
710 IF A=46H AND PO<800 THEN P1=40:GOTO 350
720 IF A<>45H THEN GOTO 350
```

Missile Command

```
5 BASE 16
1050 A=CRF[8]:B=CRF[8]:IFA<>B THEN GOTO 1130
1060 M=KEY[0]
1070 IF A=41H OR A=53H OR A=44H THEN GOSUB 1190
1080 IF A=8 AND PX>7 THEN PX=PX-4:GOTO 1120
1090 IF A=9 AND PX<240 THEN PX=PX+4:GOTO 1120
1100 IF A=0BH AND PY>8 THEN PY=PY-4:GOTO 1120
1110 IF A=0AH AND PY<152 THEN PY=PY+4
1220 IF A=53H THEN GOTO 1270
1230 IF A=44H THEN GOTO 1280
```

Note that the numbers in the comparisons are the ascii values of the keys pressed to control the games.

MDEX UTILITIES

A.R.C.Badcock

Alan has sent in a few usefull utilities for use under MDEX some of them will be included in this and future issues. But all of them, and the source listings, will be added to the utilities disk available from the Group.

XTRM Utility to tgggle between using an external terminal or the Cortex consol for main input/output. Qbasic Prog.

STRM A simmlar routine written in assembler. More compact with no messages

LCASE Lower case character loader.

MAPON Switches mapper on

MAPOFF Switches mapper off

MAPLOAD Mapper register setup utility

COLOUR Utility to set cortex screen colours

CONFIG Utility to get a readout of your MDEX I/O driver Configuration tables.

UTY.LAB Datafile for above.

All the assembler programmes are set to run at 0F200H so if you have not done the mod to get this extra memory available the source code will hav to be re-assembled to run elsware.

External terminal switch routine for mdex

A.R.C.Badcock

Written in QBASIC

```
1 {  
  EXTERNAL TERMINAL SWITCH ROUTINE  
  COPYRIGHT A.R.C.BADCOCK (C) 1984  
  
  This routine redirects console  
  I/O to an external terminal via  
  the serial port. The console  
  switch is set in the CORTEX  
  unit table pointed to by the  
  base value at 00FEh , and offset  
  by [KEYBOARD=2 and SCREEN=4].  
  Setting these words switches to  
  ext., and clearing them to int.  
}
```

STRM terminal toggle for MDEX

A.R.C.Badcock

. STRM : TERMINAL PATCH ROUTINE v1.0

. By A.R.C.Badcock - Copyright (C)1984

. This MDEX v3.3 relocatable assembler
. code is to enable switching between
. an external VDU terminal and the
. normal CORTEX console.
. The routine patches the console unit
. table at [BASE+2 & BASE+4] words.
. These are set to 0FFFFh for external
. VDU and reset to 0000h for CORTEX
. console. The BASE pointer is read
. from the guaranteed vector at 00FEh.
. The status of console path is toggled.
. command = STRM

. IDT "STRM" .Program ID
. RORG
. COPY "1/JSYS\$" .system calls
UT\$CON EQU 000FE .console table ptr
. Locate console table address
START LI R7,UT\$CON .Get pointer
MOV *R7,R8 .Get address
. Switch keyboard
KBDSET INCT R8 .Add offset
INV *R8 .Switch kbd
. Switch CRT screen
CRTSET INCT R8 .Add offset
INV *R8 .Switch crt
. Done - exit to system
FINISH JSYS FINISH
BYTE EXIT\$,0
DATA 00000
. END START . Entry label

Cortex implements its Monitor commands by matching the command letter against entries in a look-up table situated @>A3C to A90. Each entry consists of two consecutive sixteen bit words:-

Word 1. MSB is Ascii code of letter, LSB is the required number of paramemtrs.

Word 2. The entry address of the relevant routine.

There are two spare spaces in the table, but if more commands are required, these may be used to direct execution to a user written table and search routine for a two-letter command mnemonic. Thus each new entry gives up to 26 extra commands.

The ASM routine included uses 'J' as the first letter, then 'R,W,P,etc' as second letters. You may of course choose your own by changing the ascii codes accordingly.

Location @>A88 is set to >4A00, ascii 'J', with no parameters, and @>ABA to the entry address of your search routine. This could be done by a small application routine, or at CDOS boot time.

The listing was done under MDEX, but add your own start address to those in the left-hand column and enter the opcodes via the 'M' command to save under CDOS.

MON-EXT JCOM

Page 1

| | | | |
|--------|----------------|----------|--|
| 0000 | | 1. | TITL "MON-EXT" |
| 0000 | | 2. | IDT "MON COMMAND EXTENSION" |
| 0000 | 02E0 0000 | 3. JCOM | LWPI WSP1 |
| 0004 | 02A9 | 4. | STWP 9 |
| 0006 | 04CA | 5. | CLR 10 |
| 0008 | 0F46 | 6. | DATA 0F46 |
| 000A | 020B 003C | 7. | LI 11,JTAB |
| 000E | D2BB | 8. J6 | MOVB *11+,10 |
| 0010 | 1313 | 9. | JEG J5 |
| 0012 | 9286 | 10. | CB 6,10 |
| 0014 | 1303 | 11. | JEG J2 |
| 0016 | 022B 0003 | 12. | AI 11,3 |
| 001A | 10F9 | 13. | JMP J6 |
| 001C | 0F06 | 14. J2 | DATA 0F06 |
| 001E | 0009 2000 | 15. | DATA 9,02000 |
| 0022 | D2BB | 16. | MOVB *11+,10 |
| 0024 | 098A | 17. | SRL 10,8 |
| 0026 | 1306 | 18. | JEG J4 |
| 0028 | 0E46 0038 0038 | 19. J3 | DATA 0E46,J5,J5 |
| 002E | CE46 | 20. | MOV 6,*9+ |
| 0030 | 060A | 21. | DEC 10 |
| 0032 | 1BFA | 22. | JH J3 |
| 0034 | C29B | 23. J4 | MOV *11,10 |
| 0036 | 069A | 24. | BL *10 |
| 0038 | 0460 0A06 | 25. J5 | B @0A06 |
| | | 26. . | |
| 003C | 5202 0000 5702 | 27. JTAB | DATA 05202,JR,05702,JW,05000,JP |
| 0042 | 0000 5000 0000 | | |
| 0048 | 4904 0000 4F04 | 28. | DATA 04904,JI,04F04,JO,04D04,JM,04304, |
| JC,0,0 | | | |
| 004E | 0000 4D04 0000 | | |
| 0054 | 4304 0000 0000 | | |
| 005A | 0000 | | |
| 005C | | 29. | END JCOM |

These two programmes were sent in on tape by Dennis Johnson. The first one is based on othello and is designed for two players. It uses the screen for the board, pieces and score board. The second Programme is a version of the popular MASTERMIND game it only takes a few minuits to load but can get you hooked. It would be nice to see some more programmes suitable for tape sent into the newsletter so please have a look arround to see if you have anything available.

"OTHELLO" Dennis Johnson

```

1 TEXT
2 PRINT @"C"
3 GOTO 4000
5 GRAPH
7 DIM A[7,7],B[7,7],C[10]
8 I1=2: I2=2: E=0
10 GOTO 1000
15 SHAPE 1,-1,-1,-1,-1
17 SHAPE 2,0FF81H,08181H,08181H,081FFH
18 SHAPE 3,08142H,02418H,01824H,04281H
20 FOR X=28 TO 156 STEP 16
30 PLOT 60,X TO 188,X
40 NEXT X
50 FOR Y=60 TO 188 STEP 16
60 PLOT Y,28 TO Y,156
70 NEXT Y
73 SPUT 334,1: SPUT 336,2
75 SPUT 398,2: SPUT 400,1
77 B[3,3]=1: B[4,4]=1: B[3,4]=2: B[4,3]=2
80 SPRITE 1,64,31,3,15
90 H=64: V=31: P=136: D=1: X=0: Y=0
92 Z=P
100 F=KEY[0]
102 E=0
105 PRINT @(2,22);#"99";"BLACK = ";I1;#"99";" WHITE = ";I2
110 IF F=0 THEN GOTO 100
120 IF F=08H THEN GOTO 200
130 IF F=09H THEN GOTO 300
140 IF F=0AH THEN GOTO 400
150 IF F=0BH THEN GOTO 500
160 IF F=01EH THEN GOTO 600
200 IF H<=64 THEN GOTO 100
205 H=H-16: P=P-2: Y=Y-1
210 SPRITE 1,H,V
220 GOTO 100
300 IF H>=176 THEN GOTO 100
305 H=H+16: P=P+2: Y=Y+1
310 SPRITE 1,H,V
320 GOTO 100
400 IF V>=143 THEN GOTO 100
405 V=V+16: P=P+64: X=X+1
410 SPRITE 1,H,V
420 GOTO 100
500 IF V<=31 THEN GOTO 100
505 V=V-16: P=P-64: X=X-1
510 SPRITE 1,H,V
520 GOTO 100
600 IF P=Z THEN GOTO 100
602 IF B[X,Y]<>0 THEN GOTO 700

```

```

603  SPUT P,D
605  B[X,Y]=D
610  IF D=1 THEN D=2: I1=I1+1
620    ELSE D=1: I2=I2+1
630  Z=P
640  X1=X: Y1=Y
650  D2=D: IF D=1 THEN D1=2
660    ELSE D1=1
670  GOTO 2000
700  PRINT @(5,20);"ILLEGAL MOVE TRY AGAIN"
710  WAIT 100
720  PRINT @(5,20);"
730  GOTO 100
800  IF E=1 THEN GOTO 100
810  IF D=1 THEN D=2: I2=I2-1
820    ELSE D=1: I1=I1-1
830  B[X,Y]=0
840  SPUT A[X,Y],0
850  GOTO 700
1000 DATA 136,138,140,142,144,146,148,150
1010 DATA 200,202,204,206,208,210,212,214
1020 DATA 264,266,268,270,272,274,276,278
1030 DATA 328,330,332,334,336,338,340,342
1040 DATA 392,394,396,398,400,402,404,406
1050 DATA 456,458,460,462,464,466,468,470
1060 DATA 520,522,524,526,528,530,532,534
1070 DATA 584,586,588,590,592,594,596,598
1100 FOR X=0 TO 7
1110   FOR Y=0 TO 7
1130     READ A[X,Y]
1140     NEXT Y
1150   NEXT X
1160 GOTO 15
2000 Q=1: R=0
2010 IF Y1=7 THEN GOTO 2220
2020 IF Y1=6 THEN GOTO 2220
2030 M=1
2040 IF B[X1,Y1+M]=D1 THEN GOTO 2220
2045 IF B[X1,Y1+M]=0 THEN GOTO 2220
2050 C[R]=A[X,Y1+M]: R=R+1
2060 M=M+1
2070 IF B[X1,Y1+M]=D1 THEN GOTO 2110
2075 IF B[X1,Y1+M]=0 THEN GOTO 2190
2080 C[R]=A[X1,Y1+M]: R=R+1
2090 IF Y1+M=7 THEN GOTO 2190
2100 GOTO 2060
2110 R=0: M=M-1: E=1
2120 SPUT C[R],D1
2130 B[X,Y1+M]=D1
2140 IF D1=1 THEN I1=I1+1: I2=I2-1
2150   ELSE I1=I1-1: I2=I2+1
2160 R=R+1: M=M-1
2170 IF C[R]=0 THEN GOTO 2190
2180 GOTO 2120
2190 FOR J=0 TO 10
2200   C[J]=0
2210 NEXT J
2220 Q=Q+1
2230 IF Q>8 THEN GOTO 800
2240 ON Q THEN GOTO 2000,2300,2500,2700,2900,3100,3300,3500

```



```

2250 GOTO 100
2300 R=0: X1=X: Y1=Y
2310 IF Y1=7 OR X1=7 THEN GOTO 2220
2320 IF Y1=6 OR X1=6 THEN GOTO 2220
2330 M=1
2340 IF B[X1+M,Y1+M]=D1 THEN GOTO 2220
2345 IF B[X1+M,Y1+M]=0 THEN GOTO 2220
2350 C[R]=A[X1+M,Y1+M]: R=R+1
2360 M=M+1
2370 IF B[X1+M,Y1+M]=D1 THEN GOTO 2410
2375 IF B[X1+M,Y1+M]=0 THEN GOTO 2190
2380 C[R]=A[X1+M,Y1+M]: R=R+1
2390 IF Y1+M=7 THEN GOTO 2190
2395 IF X1+M=7 THEN GOTO 2190
2400 GOTO 2360
2410 R=0: M=M-1: E=1
2420 SPUT C[R],D1
2430 B[X1+M,Y1+M]=D1
2440 IF D1=1 THEN I1=I1+1: I2=I2-1
2450 ELSE I1=I1-1: I2=I2+1
2460 R=R+1: M=M-1
2470 IF C[R]=0 THEN GOTO 2190
2480 GOTO 2420
2500 R=0: X1=X: Y1=Y
2510 IF X1=7 OR X1=6 THEN GOTO 2220
2520 M=1
2540 IF B[X1+M,Y1]=D1 THEN GOTO 2220
2545 IF B[X1+M,Y1]=0 THEN GOTO 2220
2550 C[R]=A[X1+M,Y1]: R=R+1
2560 M=M+1
2570 IF B[X1+M,Y1]=D1 THEN GOTO 2610
2575 IF B[X1+M,Y1]=0 THEN GOTO 2190
2580 C[R]=A[X1+M,Y1]: R=R+1
2590 IF X1+M=7 THEN GOTO 2190
2600 GOTO 2560
2610 R=0: M=M-1: E=1
2620 SPUT C[R],D1
2630 B[X1+M,Y1]=D1
2640 IF D1=1 THEN I1=I1+1: I2=I2-1
2650 ELSE I1=I1-1: I2=I2+1
2660 R=R+1: M=M-1
2670 IF C[R]=0 THEN GOTO 2190
2680 GOTO 2620
2700 R=0: X1=X: Y1=Y
2710 IF X1=7 OR Y1=0 THEN GOTO 2220
2720 IF X1=6 OR Y1=1 THEN GOTO 2220
2730 M=1
2740 IF B[X1+M,Y1-M]=D1 THEN GOTO 2220
2745 IF B[X1+M,Y1-M]=0 THEN GOTO 2220
2750 C[R]=A[X1+M,Y1-M]: R=R+1
2760 M=M+1
2770 IF B[X1+M,Y1-M]=D1 THEN GOTO 2810
2775 IF B[X1+M,Y1-M]=0 THEN GOTO 2190
2780 C[R]=A[X1+M,Y1-M]: R=R+1
2790 IF X1+M=7 OR Y1-M=0 THEN GOTO 2190
2800 GOTO 2760
2810 R=0: M=M-1: E=1
2820 SPUT C[R],D1
2830 B[X1+M,Y1-M]=D1
2840 IF D1=1 THEN I1=I1+1: I2=I2-1

```

```

2850     ELSE I1=I1-1: I2=I2+1
2860 R=R+1: M=M-1
2870 IF C[R]=0 THEN GOTO 2190
2880 GOTO 2820
2900 R=0: X1=X: Y1=Y
2910 IF Y1=0 OR Y1=1 THEN GOTO 2220
2920 M=1
2930 IF B[X1,Y1-M]=D1 THEN GOTO 2220
2935 IF B[X1,Y1-M]=0 THEN GOTO 2220
2940 C[R]=A[X1,Y1-M]: R=R+1
2950 M=M+1
2960 IF B[X1,Y1-M]=D1 THEN GOTO 3000
2965 IF B[X1,Y1-M]=0 THEN GOTO 2190
2970 C[R]=A[X1,Y1-M]: R=R+1
2980 IF Y1-M=0 THEN GOTO 2190
2990 GOTO 2950
3000 R=0: M=M-1: E=1
3010 SPUT C[R],D1
3020 B[X1,Y1-M]=D1
3030 IF D1=1 THEN I1=I1+1: I2=I2-1
3040     ELSE I1=I1-1: I2=I2+1
3050 R=R+1: M=M-1
3060 IF C[R]=0 THEN GOTO 2190
3070 GOTO 3010
3100 R=0: X1=X: Y1=Y
3110 IF Y1=0 OR X1=0 THEN GOTO 2220
3120 IF Y1=1 OR X1=1 THEN GOTO 2220
3130 M=1
3140 IF B[X1-M,Y1-M]=D1 THEN GOTO 2220
3145 IF B[X1-M,Y1-M]=0 THEN GOTO 2220
3150 C[R]=A[X1-M,Y1-M]: R=R+1
3160 M=M+1
3170 IF B[X1-M,Y1-M]=D1 THEN GOTO 3210
3175 IF B[X1-M,Y1-M]=0 THEN GOTO 2190
3180 C[R]=A[X1-M,Y1-M]: R=R+1
3190 IF X1-M=0 OR Y1-M=0 THEN GOTO 2190
3200 GOTO 3160
3210 R=0: M=M-1: E=1
3220 SPUT C[R],D1
3230 B[X1-M,Y1-M]=D1
3240 IF D1=1 THEN I1=I1+1: I2=I2-1
3250     ELSE I1=I1-1: I2=I2+1
3260 R=R+1: M=M-1
3270 IF C[R]=0 THEN GOTO 2190
3280 GOTO 3220
3300 R=0: X1=X: Y1=Y
3310 IF X1=0 OR X1=1 THEN GOTO 2220
3330 M=1
3340 IF B[X1-M,Y1]=D1 THEN GOTO 2220
3345 IF B[X1-M,Y1]=0 THEN GOTO 2220
3350 C[R]=A[X1-M,Y1]: R=R+1
3360 M=M+1
3370 IF B[X1-M,Y1]=D1 THEN GOTO 3410
3375 IF B[X1-M,Y1]=0 THEN GOTO 2190
3380 C[R]=A[X1-M,Y1]: R=R+1
3390 IF X1-M=0 THEN GOTO 2190
3400 GOTO 3360
3410 R=0: M=M-1: E=1
3420 SPUT C[R],D1
3430 B[X1-M,Y1]=D1

```

```

3440 IF D1=1 THEN I1=I1+1: I2=I2-1
3450 ELSE I1=I1-1: I2=I2+1
3460 R=R+1: M=M-1
3470 IF C[R]=0 THEN GOTO 2190
3480 GOTO 3420
3500 R=0: X1=X: Y1=Y
3510 IF Y1=7 OR Y1=6 THEN GOTO 2220
3520 IF X1=0 OR X1=1 THEN GOTO 2220
3530 M=1
3540 IF B[X1-M,Y1+M]=D1 THEN GOTO 2220
3545 IF B[X1-M,Y1+M]=0 THEN GOTO 2220
3550 C[R]=A[X1-M,Y1+M]: R=R+1
3560 M=M+1
3570 IF B[X1-M,Y1+M]=D1 THEN GOTO 3610
3575 IF B[X1-M,Y1+M]=0 THEN GOTO 2190
3580 C[R]=A[X1-M,Y1+M]: R=R+1
3590 IF X1-M=0 OR Y1+M=7 THEN GOTO 2190
3600 GOTO 3560
3610 R=0: M=M-1: E=1
3620 SPUT C[R],D1
3630 B[X1-M,Y1+M]=D1
3640 IF D1=1 THEN I1=I1+1: I2=I2-1
3650 ELSE I1=I1-1: I2=I2+1
3660 R=R+1: M=M-1
3670 IF C[R]=0 THEN GOTO 2190
3680 GOTO 3620
4000 PRINT @"8D2R";"A BOARD GAME BASED ON OTHELLO"
4005 PRINT @"2R";"(by D.W.Johnson)"
4010 PRINT @"2R";"Moves are made with the cursor keys"
4020 PRINT @"2R";"and a piece is set with the home key."
4030 PRINT @"2R";"Black moves first and illegal moves."
4040 PRINT @"2R";"are cautioned."
4050 PRINT @"2D2R";"Press any key to start"
4060 F=KEY[0]
4070 IF F<>0 THEN GOTO 5
4080 GOTO 4060

```

```
5 PRINT @"C"
10 PRINT @"8D10R";"MASTER MIND COMPETITION"
20 PRINT @"2D10R";"GUESS THE RANDOM NUMBER"
22 PRINT @"2D13R";"(by D.W.Johnson)"
25 PRINT @"2D10R"
30 INPUT "HOW MANY DIGITS(12 MAX) ? ";D
35 PRINT @"2D10R"
40 INPUT "HOW MANY TRIES ? ";T
50 DIM R[12]: DIM S[12]: DIM B[12]: E=0
60 GOSUB 300
70 GOSUB 700
80 GOSUB 200
90 GOSUB 400
100 IF CN=D AND CP=D THEN GOTO 600
110 PRINT @"B28R";CN;: PRINT @"4R";CP
120 E=E+1
130 IF E=T THEN GOTO 500
140 GOTO 80
200 FOR A=1 TO D
210 F=KEY[0]
220 IF F=0 THEN GOTO 210
230 B[A]=F-48
235 PRINT B[A];
240 NEXT A
250 RETURN
300 FOR A=1 TO D
310 R[A]=INT[RND*9]
320 S[A]=R[A]
330 NEXT A
340 RETURN
400 CN=0: CP=0
402 FOR A=1 TO D
404 IF B[A]=S[A] THEN CP=CP+1
406 NEXT A
410 FOR A=1 TO D
420 FOR C=1 TO D
430 IF B[A]=S[C] THEN CN=CN+1: S[C]=10: B[A]=20
440 NEXT C
450 NEXT A
460 FOR C=1 TO D
470 S[C]=R[C]
480 NEXT C
490 RETURN
500 PRINT "HARD LUCK,THE NUMBER WAS"
510 FOR A=1 TO D
520 PRINT R[A];
530 NEXT A
535 PRINT
540 INPUT " TRY AGAIN ? (Y OR N)";$G
550 IF $G="Y" THEN GOTO 25
560 STOP
600 PRINT " WELL DONE"
610 GOTO 540
700 PRINT @"C"
710 PRINT @"1D1R";"YOUR ATTEMPTS"
720 PRINT @"1U28R";"RESULTS"
730 PRINT @"1D27R";"NUMBS POSNS"
740 RETURN
```

Compressed graphics screen save routine

Tim Gray

This routine copies the VDP memory into main memory using a compressed data format. Using this method most graphics screens take up less than 4K bytes instead of the 16K normal. To use the routine first reserve enough memory with

```
DIM SCN[2730]
```

Then call the routine with

```
CALL XXXX,ADR[SCN[0]]
```

Where XXXX is the start address of either the save or load routines as assembled. At the end of the save the length of the data block can be obtained from the first word of SCN[0]. So to save to disk use

```
LN=MWD[ADR[SCN[0]]]
OPEN drive,filename,filevariable,CRE,LN
PUT filevariable,0,SCN[0]
CLOSE filevariable
```

And to Recall the stored screen

```
GRAPH
OPEN drive,filename,filevariable
GET filevariable,0,SCN[0]
CALL XXXX,ADR[SCN[0]]
```

Listing Of Assembler Source File "SCNSAV.S"

```

01 ;
02 ; COMPRESSED SCREEN SAVE AND LOAD
03 ;
04 ; TIM GRAY 1988
05 ;
06 ; COMPRESSED SAVE
07 ;
08 ;
09          ORG   >F200
10 ;
11 XSVDP   EQU   >05F2
12 XVDP    EQU   >F120
13 ;
14 SAVE:   CLR   R4           ;byte count
15         CLR   R8           ;VDP start
16         MOV  R0,R9        ;save start
17         BL   @XSVDP       ;setup VDP
18         LI   R8,>4000     ;max display
19         LI   R1,>0002     ;jump count wd
20         INCT R0
21 BYTE1:  MOVB @XVDP,R2     ;get 1st byte
22         DEC  R8
23         MOVB R2,*R0+      ;save it
24         INC  R1           ;inc counter
25 NDATA:  MOVB @XVDP,R3     ;get next byte
26         DEC  R8

```

```

27          JLT  EXIT
28          CB   R2,R3          ;match ?
29          JNE  DIFF          ;no
30          INC  R4             ;yes inc by cnt
31          CI   R4,>100        ;max=256
32          JLT  NDATA         ;no next data
33          DEC  R4             ;set to 256
34  DIFF:    SLA  R4,8          ;align data
35          MOVB R4,*R0+        ;save byte count
36          CLR  R4             ;reset by count
37          MOVB R3,*R0+        ;save next data
38          MOVB R3,R2          ;copy it
39          INCT R1             ;inc counter
40          CI   R1,>4000        ;max 16K ?
41          JGT  ERROR1         ;overflow err
42          MOV  R8,R8          ;end display ?
43          JLT  EXIT          ;yes exit
44          JMP  NDATA         ;no get next
45  ERROR1:  XOP  @>0A,14       ;out of mem
46  EXIT:    MOV  R1,*R9        ;save tot count
47          SLA  R4,8          ;align data
48          MOVB R4,*R0        ;save byte count
49          RTWP
50 ;

```

PAGE 2

```

01 ;
02 ;
03 ;
04 ;COMPRESSED SCREEN LOAD
05 ;
06  LOAD:    INCT R0             ;step over count
07          LI  R8,>4000         ;write zero
08          BL  @XSVDP          ;setup VDP
09          LI  R1,>4000         ;max 16K
10  DATA:   MOVB *R0+,R2        ;get 1st data
11          MOVB *R0+,R3        ;get byte count
12          SRL  R3,8           ;align it
13  MDATA:   MOVB R2,@XVDP       ;put data in
14          DEC  R1             ;check for end
15          JEQ  EXIT1          ;yes exit
16          DEC  R3             ;byte cnt zero ?
17          JLT  DATA          ;yes get nxt dat
18          JMP  MDATA          ;no put more
19  EXIT1:   RTWP               ;end
20 ;
21 ;.      end of file
22 ;

```

Labels From Assembler File "SCNSAV.Z"

| | | | | | | | |
|-------|------|------|------|--------|------|-------|------|
| XSVDP | 05F2 | XVDP | F120 | SAVE | F200 | BYTE1 | F214 |
| NDATA | F21E | DIFF | F234 | ERROR1 | F24C | EXIT | F250 |
| LOAD | F258 | DATA | F266 | MDATA | F26C | EXIT1 | F27A |

Cortex User Group Sale Items

Hardware

| | |
|--|--------|
| R.G.B. interface P.C.B | £8.00 |
| Centronics P.C.B | £7.00 |
| E.Bus 512K DRAM P.C.B plated through hole | £40.00 |
| External Video interface P.C.B | £15.00 |
| Disk controller WD2797 + P.C.B Cortex I | £55.00 |
| Disk controller WD2797 + P.C.B Cortex II | £60.00 |
| E.Bus interface complete Kit | £30.00 |
| E.Bus 8 X 8K EPROM socket card built but no EPROMS | £30.00 |
| E.Bus 4K RAM 8K EPROM socket 16 I/O lines ex equipment | £15.00 |

| | |
|--|--------|
| TMS9902 UART IC's | £2.00 |
| 74LS612 Mapper IC's | £25.00 |
| 74LS611 or 74LS613 Mapper IC's (req pull up R's) | £10.00 |

Other IC's in stock please write in for quote

Software all disk formats please specify when ordering

| | |
|--|--------|
| CDOS basic disk system 1.20 for TMS9909 | £45.00 |
| CDOS basic disk system 2.00 for WD2797 | £45.00 |
| Wortex word processor + spelling check by J.Makenzie | £15.00 |
| Drawtech graphics drawing package by Tim Gray | £20.00 |
| Menue generator by A.R.C.Badcock | £10.00 |
| Two pass assembler by R.M.Lee | £14.00 |
| Two pass assembler by C.J.Young | £15.00 |
| Cdos utilites disk - copy charge only | £2.00 |

Cdos programmes and games all £2.50 each :-

| | | | | | |
|-----------|----------|----------|----------|----------|----------|
| ARCHIE | ASTEROID | BREAKOUT | BURGLAR | CATERPIL | C-PEDE |
| CANYON | COTELLO | FIREBIRD | FROGGER | GDESIGN | GOLF |
| HUNCHBACK | INVADERS | MAZE | MAZE-3D | MBASE | MICROPED |
| MIS-COM | MUNCHER | NIBBLERS | N-ATTACK | OLYMPICS | P.BOAT |
| PENGO | PONTOON | RESCUE | S-ATTACK | SPACE-BU | THE-ZOO |
| TRAG | VADERS | WALL | X/O | | |

MDEX Software all formats please specify

| | |
|--|--------|
| MDEX CORE with debug monitor text editor and basic | £10.00 |
| MDEX ASM & LINK assembler and linker | £10.00 |
| MDEX SYSGEN system generation kit | £10.00 |
| MDEX WORD word processor | £10.00 |
| MDEX P.D.S. all the above in one package | £30.00 |
| MDEX S.P.L. system programming language | £10.00 |
| MDEX META compiler generator | £10.00 |
| MDEX QBASIC basic compiler | £15.00 |
| MDEX PASCAL sequential pascal | £10.00 |
| MDEX WINDOW full screen editor | £15.00 |
| MDEX SPELL spelling checker | £10.00 |
| MDEX utilities copy charge only | £2.00 |